

(Optional) Advanced MVC in Laravel

Clean Request Handling

Advanced Routing Features

Route Groups:

```
Route::prefix('api/v1')->group(function () {  
    Route::apiResource('students', StudentController::class);  
    Route::apiResource('courses', CourseController::class);  
});  
// Creates: /api/v1/students, /api/v1/courses
```

The Problem

- Large projects have **many controllers & endpoints**
- Without organization:
 - URLs become inconsistent
 - Versioning is hard (/students , /courses , /users , ...)
 - No clear separation between **web routes** and **API routes**

Example Without Prefix

```
/students  
/courses  
/users
```

Problems:

- Hard to know if it's API v1, v2, or web routes
- Migration to a new API version requires renaming every route manually
- Potential conflicts with web pages (/students view vs /students API)

Route Prefix Solution

```
Route::prefix('api/v1')->group(function () {  
    Route::apiResource('students', StudentController::class);  
    Route::apiResource('courses', CourseController::class);  
});  
// API v2 → New version  
Route::prefix('api/v2')->group(function () {  
    Route::apiResource('students', StudentV2Controller::class);  
});
```

Laravel automatically prepends /api/v1 to all routes inside the group, so all routes inside become:

```
/api/v1/students  
/api/v1/courses  
/api/v2/students
```

Big Picture

[Controllers] → Students, Courses



[Route::apiResource] → Generates CRUD routes



[Route::prefix('api/v1')] → Adds versioning & organization



Final Routes: /api/v1/students, /api/v1/courses

Route Middleware:

```
Route::middleware('auth:api')->group(function () {  
    Route::apiResource('students', StudentController::class);  
});  
// Requires authentication for all student routes
```

The Problem

- By default, anyone can hit your API routes if they know the URL.
- We need a way to protect sensitive routes so only authenticated users (with a valid token) can access them.

The Solution → `middleware('auth:api')`

- Middleware = code that runs before the request reaches the controller.
- `auth:api` is a middleware provided by Laravel.
- It checks:
 - i. Does the request include a valid API token (e.g., Bearer token)?
 - ii. If yes → continue to the controller.
 - iii. If no → return 401 Unauthorized.

Parameter Validation

Route Constraints:

```
Route::get('/students/{id}', [StudentController::class, 'show'])  
    ->where('id', '[0-9]+'); // Only accept numbers
```

Controller Validation:

```
public function store(Request $request)
{
    $request->validate([
        'name' => 'required|string|max:255',
        'email' => 'required|email|unique:students',
        'year' => 'required|integer|min:1|max:4'
    ]);

    // If validation fails, Laravel automatically returns an error
    $student = Student::create($request->all());
    return response()->json($student, 201);
}
```

Error Handling

PHP Manual Way:

```
function get_student($id) {  
    $students = load_students();  
    foreach ($students as $student) {  
        if ($student['id'] == $id) {  
            echo json_encode(['success' => true, 'data' => $student]);  
            return;  
        }  
    }  
    http_response_code(404);  
    echo json_encode(['success' => false, 'error' => 'Student not found']);  
}
```

Laravel Way:

```
public function show($id)
{
    $student = Student::findOrFail($id); // Auto 404 if not found
    return response()->json($student);
}
```

How Laravel Distinguishes `web.php` vs `api.php`

Two Different Route Files

- `routes/web.php` → for **web routes**
- `routes/api.php` → for **API routes**

Both can technically have the same path like `/students`.

Middleware Groups

- `web.php` routes are loaded with the `web` middleware group:
 - Handles **sessions, cookies, CSRF, views**
- `api.php` routes are loaded with the `api` middleware group:
 - **Stateless**, no sessions, no CSRF
 - Automatically prefixed with `/api`

Automatic Prefix for API Routes

If you define this in `api.php`:

```
Route::get('/students', [StudentController::class, 'index']);
```

What Laravel creates: `/api/students`

```
Full URL: http://localhost:8000/api/students
```

Middleware Group Applied

Laravel automatically wraps your route with the api middleware group:

Laravel internally does this:

```
Route::middleware(['api'])->group(function () {  
    Route::get('/students', [StudentController::class, 'index']);  
});
```